Fixing the Sequential Bottleneck by Regulating Energy Per Instruction on CMPs

Murali Annavaram
Senior Researcher
Microarchitecture Research Lab
Intel Corporation

Ed Grochowski
Senior Principal Engineer
Microarchitecture Research Lab
Intel Corporation

John Shen
Director
Microarchitecture Research Lab
Intel Corporation

Table of Contents

(Click on page number to jump to sections)

FIXING THE SEQUENTIAL BOTTLENECK BY REGULATING ENERGY PER INSTRUCTION ON CMPS

- Overview: Mitigating Amdahl's Law 3
- Tough Challenges 3
- Prototype AMP Configuration 4
- Staying Within the Power Budget 4
- Workloads 5
- Performance Results 5
- Advantages of AMP 5
- Practical EPI Throttles 5
- Summary 6
- More Info 6
- Author Bios 7

Fixing the Sequential Bottleneck by Regulating Energy Per Instruction on CMPs

Murali Annavaram
Senior Researcher
Microarchitecture Research Lab
Intel Corporation

Ed Grochowski
Senior Principal Engineer
Microarchitecture Research Lab
Intel Corporation

John Shen
Director
Microarchitecture Research Lab
Intel Corporation

Overview: Mitigating Amdahl's Law

The world is becoming more parallel. With the acceptance of dual-core microprocessors and the advent of chip multiprocessors (CMPs) that can contain a large number of processors, developers are faced with the challenges of code parallelization. Similarly, chip designers are faced with the challenges of designing CMPs that fit within a constrained power budget, and yet which can efficiently run both the sequential and parallel phases of a program.

The key to designing a CMP that can achieve both high scalar (sequential) performance and high throughput (multithreaded) performance is to dynamically vary the EPI (energy per instruction, or the amount of energy expended to process each instruction), according to the amount of parallelism available in the software. In other words, if there is limited parallelism, a microprocessor should expend all available energy processing a few instructions. If there is a great deal of parallelism, the microprocessor should expend less energy in processing each instruction.

Now, for the first time, Intel researchers have experimentally demonstrated the effectiveness of EPI throttling using an asymmetric multiprocessor (AMP) prototype. The results of these tests are dramatic. The prototype AMP, with its novel combination of clock throttling and processor affinity, shows a 38 percent wall clock speed-up for a wide range of multithreaded programs, as compared to a symmetric multiprocessor (SMP) that uses the same power.

These dramatic results make a compelling case that EPI throttling is effective in mitigating the effects of Amdahl's law, and is expected to become an important aspect of future CMP designs.

Tough Challenges

There are three main challenges to improving CMP system performance.

First is mitigating the effects of Amdahl's law, which states that the performance gain, or speed-up, seen from code parallelization will be limited by the amount of time the system spends processing the sequential part of the code. For example, if a program spends 10 percent of its time in a sequential component, the maximum speed-up achievable through parallelization is 10. Since almost all multithreaded programs will still contain some, even if not significant amounts of sequential code, it is critical to improve sequential execution.

The second challenge is improving performance within the constraints of a power budget. Historically, chip designers have extended the projections of Moore's Law for 40 years, dramatically reducing transistor size while simultaneously reducing transistor power consumption. However, designers are now coming up against the physical, atomic limitations of today's materials science. Advances in power technology are now lagging behind advances in transistor technology, making power/thermal issues an increasingly critical design (and performance) constraint.

The third challenge is satisfying the conflicting microarchitectural demands of parallel vs. sequential processing. It is almost impossible to optimize a processor for both single-threaded (sequential) and throughput (parallel, or multithreaded) performance within a fixed power budget. For example, most of today's microarchitectural techniques (such as out-of-order execution, speculation, deep pipelining, etc.) help reduce latency only for single-threaded execution. They also consume relatively high energy per instruction, or EPI. That high EPI limits the number of CPU cores that can be accommodated in a CMP for a given power budget. On the other hand, throughput performance demands many low-energy cores on a CMP in order to exploit thread-level parallelism.

To help resolve these challenges, Intel researchers have been exploring EPI throttling to improve both scalar and throughput performance on power-constrained designs.

Prototype AMP Configuration

To evaluate the performance benefits of EPI throttling, researchers constructed a prototype, asymmetric multiprocessor (AMP) system with an off-the-shelf, Intel® Xeon™ processor-based 4-way SMP server as the base machine. The prototype uses a novel combination of the Intel® Pentium® 4 processor's clock throttle mechanism along with processor affinity, to create multiple performance and power operating points. In the prototype system, each processor expends a varying amount of energy per retired instruction, based on the available thread-level parallelism.

Developers should note that the physical AMP prototype system and the software modifications (described in detail in the research paper "Mitigating Amdahl's Law through EPI Throttling") were primarily intended for evaluating the benefits of EPI throttling. They were not intended to provide a definitive hardware implementation. In practice, an EPI throttle may be implemented as a hardware mechanism that operates transparently to software.

The prototype used a conservative square relationship between power and performance. The power budget was fixed to be the same as the power consumed by a 1P Xeon processor running at 2 GHz. Given this power budget, the base SMP can be configured into several configurations using different duty cycles (refer to **Table 1**), all of which consume roughly the same power, based on the square relationship between power and performance.

CPUs	Duty Cycle	Effective Frequency	Normalized Total Power
1P	8/8	2 GHz	1.00 (Baseline)
2P	6/8	1.5 GHz	1.12
3P	5/8	1.25 GHz	1.17
4P	4/8	1 GHz	1.00

Table 1. *MP Configurations with Similar Power*

Processor affinity was used to tell the operating-system scheduler how to assign sequential and parallel processes to a particular CPU. In the tests, parallel phases of code execution were assigned to the low-EPI processors. Sequential phases were assigned to the high-EPI processors. This meant that sequential code was assigned (for example) to the fast, 2-GHz processor, while parallel code was assigned to run as parallel threads on three slower, 1.25-GHz processors.

Staying Within the Power Budget

Although clock throttling does not reduce the actual operating voltage or frequency of a processor, it does emulate the behavior of an EPI throttle.

For example, in one test, when sequential code was executed, only a single 2-GHz processor was used, and the remaining three processors were assumed to be shut off so that they consumed negligible power. Similarly, when parallel code with four parallel threads was executed, all four processors were run at 1 GHz, with a duty cycle of 4/8. When execution was reduced to two threads, the threads were assigned to just two processors, which were run at 1.5-GHz, with a duty cycle of 6/8.

In essence, the prototype used the amount of thread-level parallelism in the program to determine the number of processors to use, and to figure the corresponding frequency (duty cycle) to use to stay within the power budget. Since EPI is lower during parallel execution phases and EPI is higher during sequential execution phases, this approach allowed the prototype AMP to consume a constant level of power throughout program execution.

Workloads

Tests were run on a range of realistic, multithreaded programs, such as bioinformatics programs, decision-support programs, and a parallel Fourier-transform solver. The programs offered different ratios of sequential-to-parallel execution. For example, in one bioinformatics program, 30 percent of the execution time was spent in database formatting (a sequential operation), while 70 percent of the execution time was spent in search operations (parallel operations).

Intel researchers also explored whether a static or dynamic AMP was more efficient. They found that, when programs shift rapidly between sequential and parallel phases, there is significant overhead in frequently changing the duty cycle. In these cases, a static AMP may be more efficient.

On the other hand, when thread-level parallelism varies and there are only a few shifts (thread migration) between sequential and parallel phases, there is less overhead for the changes in duty cycle. In this case, a dynamic AMP was more efficient.

Performance Results

Based on the wall clock speed-ups achieved during program execution, the programs used in the tests fell into three categories:

- *Highly parallel, CPU-intensive programs.* In today's environment, these programs perform slightly better on multiple, low-power processors, because they don't execute enough sequential code to benefit significantly from EPI throttling.
- *Moderately parallel programs.* These programs spent about 23 percent to 36 percent of their execution time in sequential processing. They saw a significant benefit from EPI throttling on the prototype AMP, as compared to execution on either a single-CPU or a standard SMP.
- *Highly sequential programs.* While these programs spent about 31 percent to 54 percent of execution time in sequential processing, they also shifted rapidly between sequential and parallel processing. The duty-cycle overhead inherent in such shifts offset any reduction in execution time offered by the AMP, and these programs saw no performance benefit on the AMP than on a standard 2-GHz processor.

Advantages of AMP

The prototype AMP was able to offer its dramatic speed-up in wall clock performance because of the limitations inherent in non-EPI systems.

Always using multiple processors gives you a configuration optimized for parallel processing, but not efficient for highly sequential code. For example, when code is 100 percent sequential, a four-processor, 1-GHz system takes twice the execution time that a one-processor 2-GHz configuration does, since only one of the four CPUs is used to execute all that sequential code. Also, the power budget is underutilized because only one of the four processors is running during sequential execution.

On the other hand, always using a single, faster (2 GHz) processor means that the system doesn't take advantage of available thread-level parallelism during parallel phases of execution. For example, when code is fully multithreaded, a single 2-GHz processor takes twice the execution time compared to an SMP system with four 1-GHz processors, because it executes all that multithreaded code in sequence.

Practical EPI Throttles

Researchers expect that, in practical implementations, an EPI throttle will monitor the activity levels of the various blocks within a chip. Based on that activity, the throttle would then determine when to reassign programs to particular CPUs and when to reduce the power consumption.

One obvious question is how well will EPI throttling perform on future CMP systems and software? As the number of processors increases, researchers expect that the performance benefits of EPI throttling will increase. This is due to Amdahl's law: As the parallel phase is divided among more and more CPUs, it becomes increasingly important to run the sequential phase quickly. Multiprocessors that can vary EPI over a wide range will have a performance advantage over multiprocessors that run at a nearly constant EPI.

Summary

There is a shift in the world toward parallel computing, with a corresponding shift in the way software gets written—a shift that is beginning to become apparent with today's dual-core processors, and one that will become more important as the world moves to CMPs. Unfortunately, as processors become more powerful to handle multithreading technology, their power/thermal technologies become increasingly challenging to control. To move forward again, designers must find new ways of achieving performance gains through better design, while staying within a given power budget.

The good news is that Intel researchers have found a novel way to take advantage of a variety of today's technologies—CMPs, multi-core processors, multithreading technology, and flexible power-down techniques—to dramatically improve performance within a fixed power budget.

By using a unique combination of clock throttling and processor affinity, researchers have shown that an EPI-throttled asymmetric multiprocessor can dramatically improve performance. This physical AMP prototype has demonstrated an average wall-clock speed-up of 38 percent, significantly mitigating the effects of Amdahl's law, while running nontrivial amounts of both sequential and parallel code.

In the future, EPI throttles may even be able to operate under several competing constraints, such as minimizing energy, minimizing di/dt-induced supply voltage variation, reducing the magnitude of thermal hot spots, or guaranteeing a certain quality of service.

Intel researchers believe EPI (energy per instruction) throttling will be an essential aspect of future chip multiprocessors, and intend to continue exploring the promise of this approach to improve both power utilization and performance.

More Info

Detailed information about Intel's research into a physical EPI throttle can be found in the paper titled, "Mitigating Amdahl's Law through EPI Throttling," on the University of Wisconsin Madison, Computer Science Web site. The paper includes information about related research, test goals and results, the prototype configuration, software simulations, and a possible hardware implementation for a software-transparent EPI throttle.

For information about possible ways to build an EPI throttle, refer to the paper titled "Best of Both Latency and Throughput," on the Web site of the IEEE International Conference on Computer Design.

Explore the Intel Technology & Research Web site for much more information about Intel's advanced research.

Learn more about Amdahl's law.

References

Original Conference Paper: M. Annavaram, E. Grochowski, J. Shen. "Mitigating Amdahl's Law through EPI Throttling." In *Proceedings of the 32nd International Symposium on Computer Architecture*, pages 298-309, June 2005

Author Bios

Murali Annavaram is a senior researcher at the Microarchitecture Research Lab in Austin, Texas. His current research focuses on squeezing more performance from microprocessors without sacrificing power. In the past he has worked on server workload characterization, trace analysis and full-system simulation infrastructure for server workloads. Prior to joining Intel he spent the best part of his youth at the University of Michigan working with Prof. Ed Davidson on prefetching techniques for databases. He earned his Ph.D. in computer science from the University of Michigan. Murali has four patents pending in microarchitecture and variation-tolerant designs.

Ed Grochowski is a senior principal engineer at the Microarchitecture Research Lab in Santa Clara, California. He joined Intel in 1986 and has had various technical and managerial responsibilities in the Intel® i486™, Pentium®, Pentium® II, and Itanium® microprocessor design teams. Grochowski currently works on the microarchitectural techniques needed for future energy-efficient chip-level multiprocessors. He received both his B.S.E.E. and his M.S.E.E. from the University of California, Berkeley. Grochowski holds over 30 patents in the areas of microarchitecture and logic design.

John P. Shen is currently the director of the Microarchitecture Research Lab at Intel. His team, located in Santa Clara, California; Hillsboro, Oregon; and Austin, Texas, is responsible for developing and evaluating innovative microarchitecture techniques that can potentially be adopted by the design teams of future Intel IA-32 and Intel® Itanium® microprocessors. Prior to joining Intel in 2000, Shen was a full professor in the E.C.E. Department of Carnegie Mellon University. You can learn more by reading an interview with John Shen.

—End of Technology@Intel Magazine Article—